AD-774 072

INTRODUCTION TO FAULT TREE ANALYSIS

Richard E. Barlow, et al

California University

Prepared for:

Office of Naval Research
Air Force Systems Command

December 1973

INTRODUCTION TO FAULT TREE ANALYSIS

by

Richard E. Barlow
Department of Industrial Engineering
and Operations Research
University of California, Berkeley

and

Purnendu Chatterjee
Operations Research Center
University of California, Berkeley

DECEMBER 1973 ORC 73-30

This research has been partially supported by the Office of Naval Research under Contract N00014-69-A-0200-1036 and the United States Air Force under Contract F33615-73-C-4078 with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

48

1.

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) University of California, Berkeley	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

INTRODUCTION TO FAULT TREE ANALYSIS

4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)
Research Report

5. AUTHOR(S) (First name, middle initial, last name)

Richard E. Barlow and Purnendu Chatterjee

6. REPORT DATE December 1973	7a. TOTAL NO. OF PAGES 42	7b. NO. OF REFS 13

8a. CONTRACT OR GRANT NO. F33615-73-C-4078 b. PROJECT NO. 7071 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) ORC 73-30 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES Also supported by the Office of Naval Research under Contract N00014-69-A-0200-1036.	12. SPONSORING MILITARY ACTIVITY United States Air Force Air Force Systems Command 4950th Test Wing/PMMA Wright-Patterson Air Force Base, Ohio 45433

13. ABSTRACT

SEE ABSTRACT.

Unclassified
Security Classification

Unclassified

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Fault Tree Analysis						
Min Cut Sets						
Dual Events and Trees						
Binary Indicated Cut Sets						
Association						
Importance						
Coherent Structures						

ABSTRACT

Fault tree analysis has proved to be a useful analytical tool for the reliability and safety analysis of complex systems. This is a semi-expository introduction to the mathematics of fault tree analysis. Many of the concepts of coherent structure theory have been used. Bounds on the system reliability when components are dependent (that is, are associated) are given. Algorithms to find the min-cut-sets and related bounds, together with various means for computing the probability of the Top Event are presented. Measures of event importance are discussed. Numerical examples are presented to illustrate the concepts.

CONFERENCE ON RELIABILITY AND FAULT TREE ANALYSIS

The purpose of the conference is to bring forth recent developments in the fields of Reliability and Fault Tree Analysis. Models, concepts and methods of quantitative analysis in both fields have significant similarities. This conference, for the first time, will provide a ground for the exchange of ideas among various groups that have been following independent paths.

SESSIONS WILL INCLUDE:

1. Fault Tree Construction
2. Fault Tree Analysis
3. Coherent Structures and Combinatorics
4. Statistical Problems in Reliability and Fault Tree Analysis
5. Network Reliability
6. Computer Reliability
7. Applications to Nuclear Power Reactors and Other Fields

Apart from these sessions, an organized panel discussion will be scheduled to bring forth unsolved but crucial problems in these areas.

The conference will be held at the University of California, Berkeley Campus, from September 3 to September 7, 1974. In the unique setting of the San Francisco Bay Area, the conference will provide both professional excellence and a taste of cosmopolitan living.

The registration fee is $25.00. For further information, contact P. Chatterjee, Operations Research Center, University of California, Berkeley, California 94720 (415) 642-4993.

Program Committee:

R. E. Barlow
P. Chatterjee

INTRODUCTION TO FAULT TREE ANALYSIS

by

Richard E. Barlow and Purnendu Chatterjee

0. INTRODUCTION AND SUMMARY

This is a semi-expository introduction to the mathematics of fault tree analysis. The literature on fault tree analysis is, for the most part, scattered through conference proceedings and company reports. Therefore, we feel that a readable, logical introduction to the subject is very much needed. A discussion of fault tree construction may be found in Lambert (1973). A description of fault tree concepts and techniques can also be found in Fussell (1973). Vesely (1970) has considered fault tree analysis from the point of view of computer implementation.

Our main contribution is to develop a mathematical theory of fault tree analysis using many of the concepts of coherent structure theory [Birnbaum, Esary and Saunders (1961)] and to show how dependent events may be analyzed.

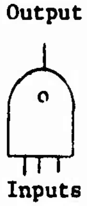
It has been observed by reliability theorists that many of the quantities computed by fault tree analysts can also be computed using the concepts and techniques of reliability theory. While this is true, we feel that the tree structure used by fault tree analysts and the somewhat different problems of interest to fault tree analysts, warrant a separate development.

In Section 1 we present some examples of fault trees and the symbols used. In Section 2 we describe some algorithms due to J. Fussell (1973) for analyzing fault trees. Dual fault trees and their uses are described in Section 3. Section 4 is a lengthy development of methods for probability evaluation of fault trees. New results on computing probabilities for trees with dependent events are presented. Section 5 considers measures of event importance. Many concepts are illustrated using the pressure tank example introduced in Section 1.

1. FAULT TREES

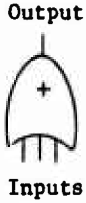
To construct fault trees we employ the following useful symbolism. Component states or, more generally, *basic events* will be represented by circles and diamonds. A system event of major importance will be represented by a rectangle

called the *Top Event*, appearing at the top of the fault tree. For example, this may indicate a particular type of system failure. *Intermediate* system or subsystem events will also be represented by rectangles. Immediately below each rectangle will be either an AND gate represented by



AND GATE

or an OR gate represented by



OR GATE

The output event to an AND gate occurs if and only if all input events occur. It is helpful to put a dot (for set product or intersection) in the center of the AND gate. For example, to symbolize that if each of the events A , B , C , and D occur, then the event E will occur, the fault tree analyst would draw

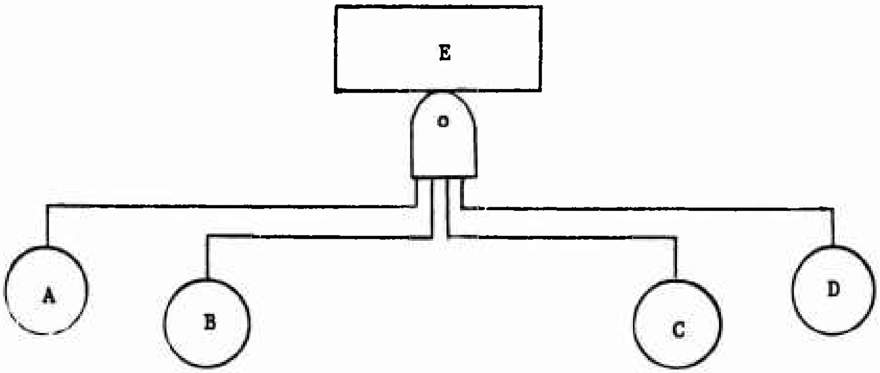


FIGURE 1: AND GATE

The output event to an OR gate occurs if one or more of the input events occur. It is helpful to put a plus (for set sum or union) in the center of the OR gate. For example, E occurs if one or more of the events A , B , C or D occurs in Figure 2.

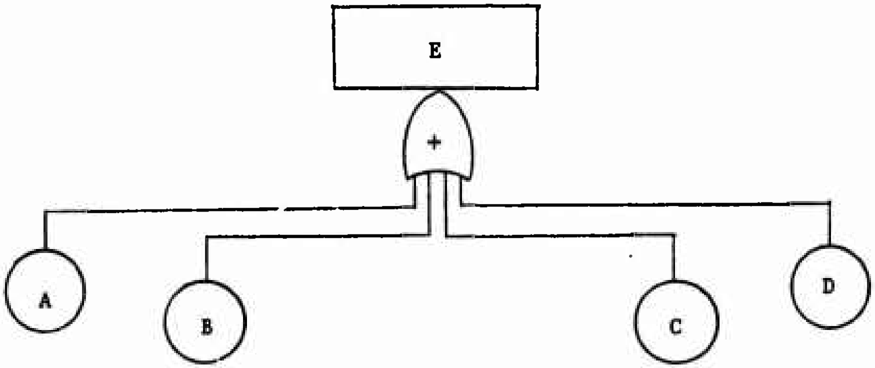


FIGURE 2: OR GATE

Repetition of basic events is permitted in a fault tree.

Example. One-out-of-Two Twice System.

Figure 3 symbolizes a system whose function is to shut down a nuclear power plant in the event of a low coolant pressure. The 2-out-of-2 coincidence unit produces a trip signal provided that the "OR" unit in both the upper and lower branches simultaneously produces an output signal. Such logic is called *one-out-of-two twice*. Units c_1 through c_4 are pressure switches. The i^{th} switch will produce an output signal (we call this basic event i) if the pressure p_i drops below a prescribed value, $i = 1, \ldots, 4$. A fault tree for this system with Top Event, "Spurious Trip Signal Produced," is shown in Figure 4.

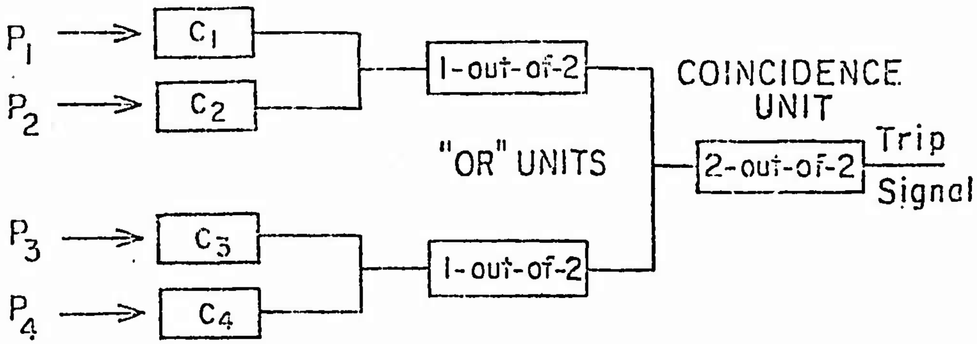


FIGURE 3: ONE-OUT-OF-TWO TWICE SYSTEM

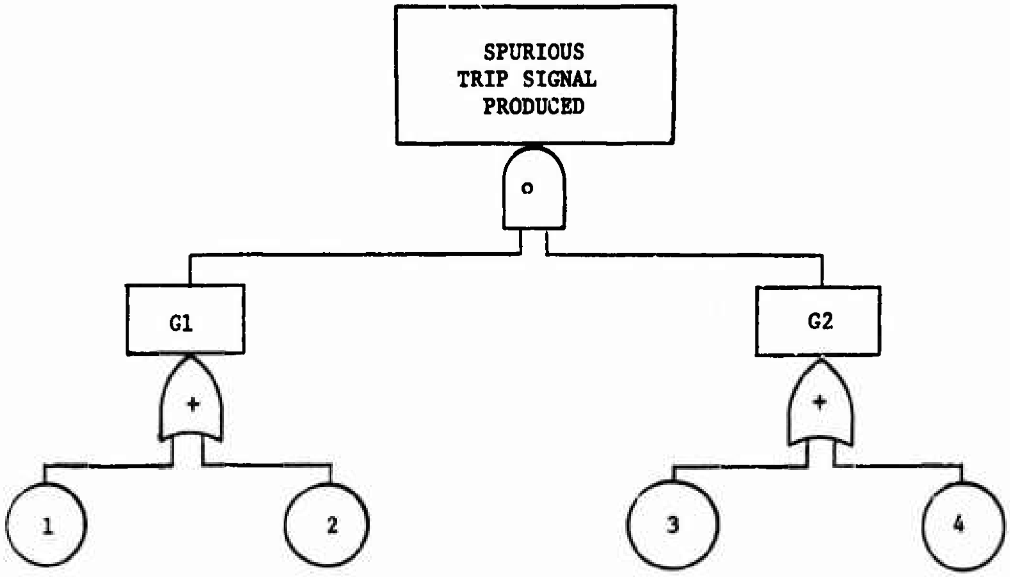


FIGURE 4: FAULT TREE FOR ONE-OUT-OF-TWO TWICE SYSTEM

Example. Pressure Tank System.

Consider the pressure tank engineering diagram in Figure 5. Let the top event (which we wish to prevent) be the rupture of the pressure tank. To start pumping, the switch S1 (a push button) is closed and then immediately opened. This allows current to flow in the control branch circuit which activates relay coil K2 . Relay contacts K2 then close and start up the pump motor. After a period of approximately 20 seconds, the pressure switch contacts open (since excess pressure is detected by the pressure switch), deactivating the control circuit which de-energizes the K2 coil. The K2 contacts then open and shut off the motor. If there is a pressure switch malfunction, then the timer relay contacts open after 60 seconds, de-energizing coil K2 , and shutting off the pump. The timer resets itself automatically after each cycle.

The fault tree drawn in Figure 6 is based on an analysis of the possible failure modes of the system. Circles represent *primary basic events*, while diamonds represent *secondary basic events*. For example, if the K1 relay contacts (Figure 5) fail to open under normal operating conditions (i.e., within the "design envelope"), this is considered a primary basic event. If the K1 relay fails to open because the wrong relay was installed, then this is considered a secondary basic event. A systematic method for drawing fault trees has been developed by David Haasl (1965). The pressure tank example is due to Haasl [cf. also Lambert (1973)].

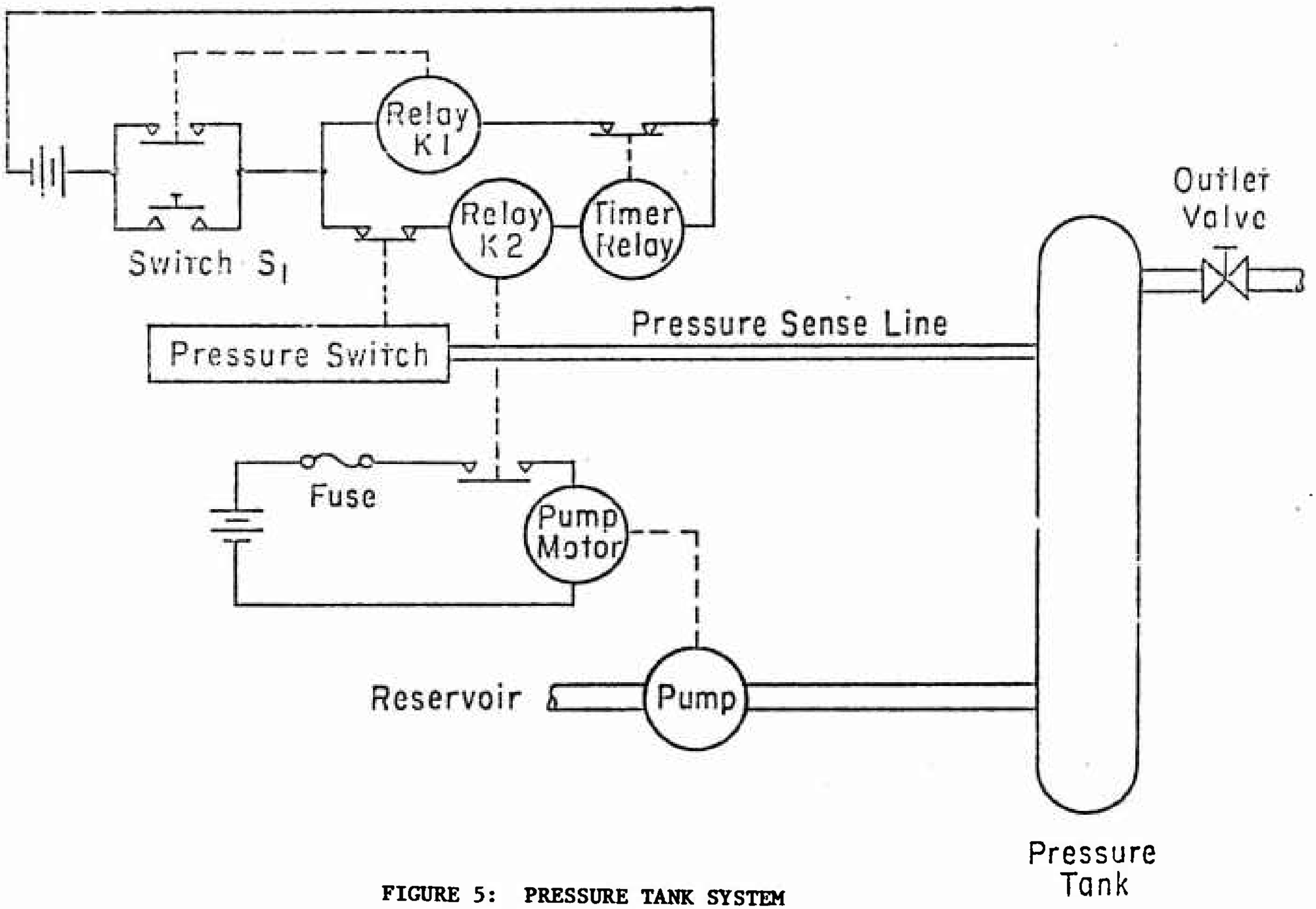


FIGURE 5: PRESSURE TANK SYSTEM

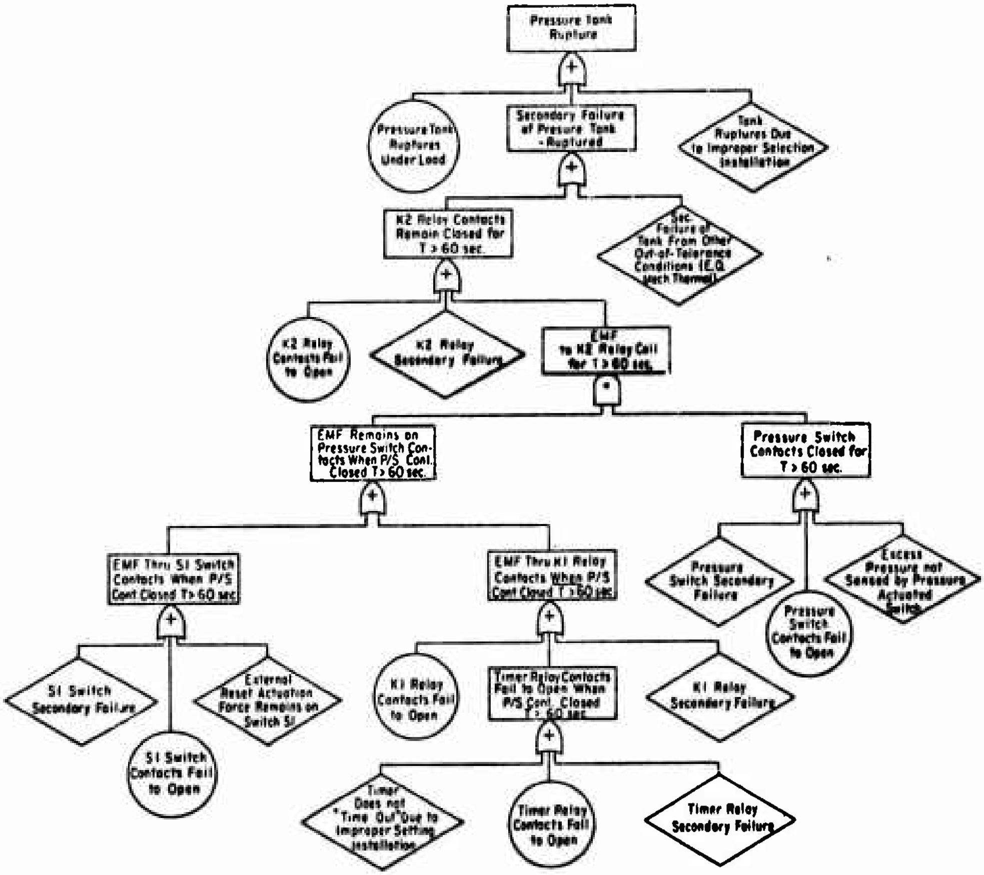

Pressure Tank Rupture
+
Pressure Tank Ruptures Under Load
Secondary Failure of Presure Tank - Ruptured
Tank Ruptures Due to Improper Selection installation
+
K2 Relay Contacts Remain Closed for T > 60 sec.
Sec. Failure of Tank From Other Out-of-Tolerance Conditions (E.G. Mech Thermal)
+
K2 Relay Contacts Fail to Open
K2 Relay Secondary Failure
EMF to K2 Relay Coil for T > 60 sec.
•
EMF Remains on Pressure Switch Contacts When P/S Cont. Closed T > 60 sec.
Pressure Switch Contacts Closed for T > 60 sec.
+
+
EMF Thru S1 Switch Contacts When P/S Cont Closed T > 60 sec
EMF Thru K1 Relay Contacts When P/S Cont Closed T > 60 sec
Pressure Switch Secondary Failure
Pressure Switch Contacts Fail to Open
Excess Pressure not Sensed by Pressure Actuated Switch
+
+
S1 Switch Secondary Failure
S1 Switch Contacts Fail to Open
External Reset Actuation Force Remains on Switch S1
K1 Relay Contacts Fail to Open
Timer Relay Contacts Fail to Open When P/S Cont. Closed T > 60 sec
K1 Relay Secondary Failure
+
Timer Does not "Time Out" Due to Improper Setting installation
Timer Relay Contacts Fail to Open
Timer Relay Secondary Failure

FIGURE 6: PRESSURE TANK FAULT TREE

Generally, fault trees serve three purposes:

1. In safety analysis, a fault tree aids in determining the possible causes of an accident. When properly used, the fault tree often leads to discovery of failure combinations which otherwise might not have been recognized as causes of the event being analyzed.
2. The fault tree serves as a display of results. If the system design is not adequate, the fault tree can be used to show what the weak points are and how they lead to undesirable events. If the design is adequate, the fault tree can be used to show that all conceivable causes have been considered.
3. The fault tree provides a convenient and efficient format helpful in the computation of the probability of system failure.

2. MINIMUM CUT SET ALGORITHM

A *cut set* is a set of basic events whose occurrence causes the Top Event to occur. A cut set is *minimal* if it cannot be reduced and still insure the occurrence of the top event. A listing of minimal cut sets (or *min cut sets*) is useful for design purposes in order to determine the "weakest links" in the system.

For a fault tree with perhaps hundreds of gates and hundreds of basic events it is clearly not easy, nor in general possible, to determine all min cut sets by inspection. An algorithm is therefore required to generate all min cut sets. The algorithm is based on the fact that an AND gate always increases the *size* of a cut set while an OR gate always increases the *number* of cut sets. The algorithm obtains cut sets such that, if all the primary events were different, the cut sets so generated would be precisely the minimal cut sets. When this is not the case, the cut sets generated by the algorithm are then reduced to minimal cut sets. This algorithm was first stated by J. Fussell and W. Vesely (1972).

The simplest and clearest way to explain the *min cut set algorithm* is to illustrate its operation in an example. Figure 7 is a relabelling of the basic events and gates in the pressure tank fault tree described in Figure 6. AND and OR gates are labelled G-1 through G-8. The algorithm begins with the gate immediately below the top event, which we label G-0. If G-0 is an OR gate, each input is used as an entry in separate rows of a list matrix. If G-0 is an AND gate, each input is used as an entry in the first row of a list matrix. Since in Figure 9, the gate immediately below the top event is an OR gate we begin the construction of our list matrix by listing inputs 1 , G-1 , and 2 in separate rows as follows:

1

G-1

2

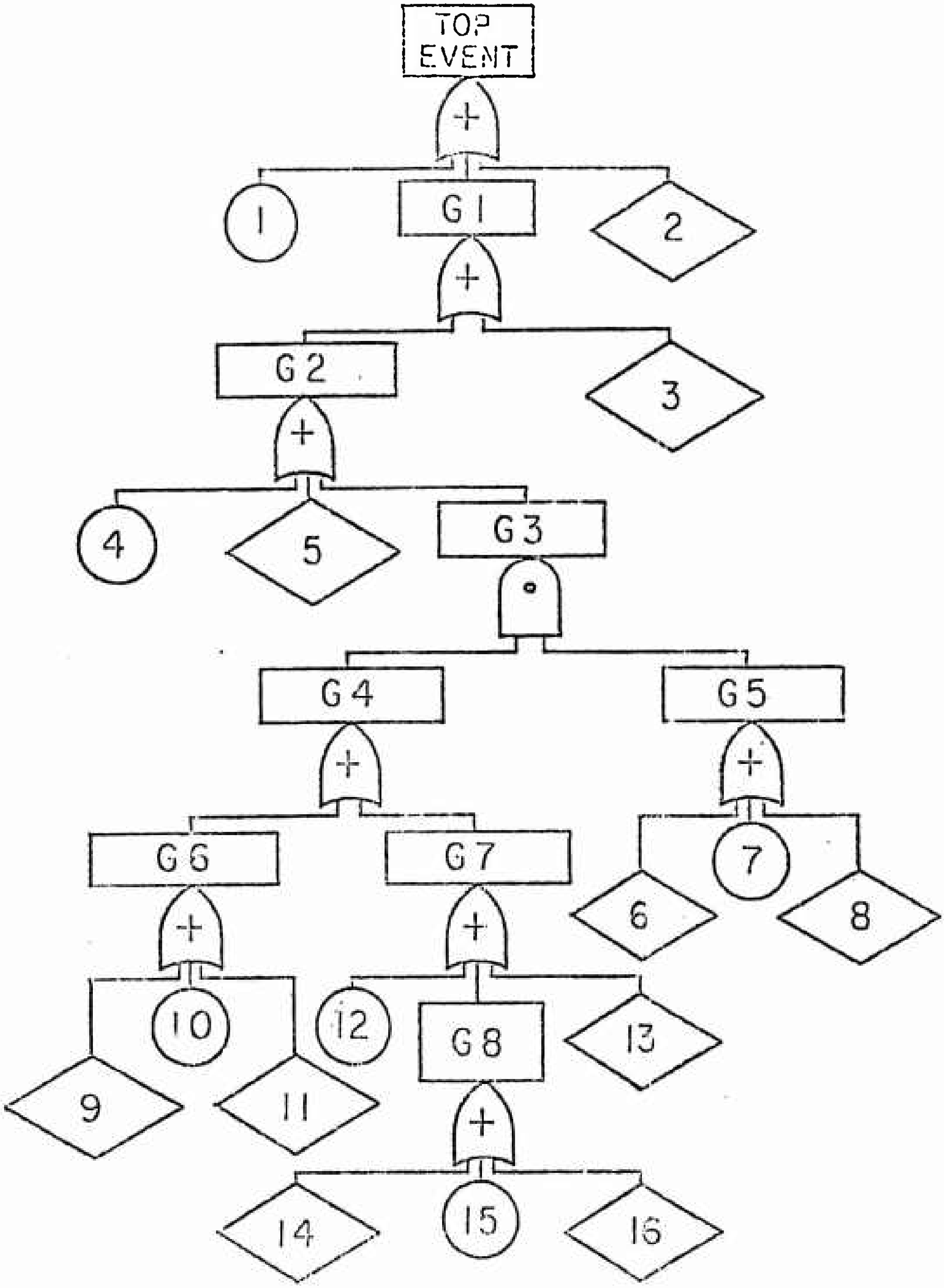


FIGURE 7: PRESSURE TANK FAULT TREE

Since any one of these input events can cause the top event to occur, each will be a member of a separate cut set.

The idea of the algorithm is to replace each gate by its input gates and basic events until a list matrix is constructed, all of whose entries are basic events. The rows will then correspond to cut sets.

Since G-1 is an OR gate, we again replace G-1 by its input events in separate rows as follows:

1

G-2

3

2 .

Since G-2 is also an OR gate, we replace G-2 by its input events as follows:

1

4

5

G-3

3

2 .

Since G-3 is an AND gate, we replace the *row* containing G-3 by its inputs as follows:

1

4

5

G-4 , G - 5

3

2 .

Since all inputs to an AND gate must occur to cause the corresponding intermediate event above the AND gate, we see that an AND gate increases the length of its row. An OR gate, on the other hand, increases the number of rows in our list matrix.

Replacing G-4 by its inputs, we have

1
4
5
G-6 , G-5
G-7 , G-5
3
2 .

Continuing in this fashion we eventually obtain a list matrix with 29 rows. These are (in a different order),

1
2
3
4
5
6 , 9
6 , 10
6 , 11
6 , 12
6 , 13
6 , 14
6 , 15
6 , 16

7 , 9
7 , 10
7 , 11
7 , 12
7 , 13
7 , 14
7 , 15
7 , 16

8 , 9
8 , 10
8 , 11
8 , 12
8 , 13
8 , 14
8 , 15
8 , 16

In the pressure tank fault tree (Figure 7), basic events are *not* repeated. For this reason all of our cut sets are minimal cut sets; i.e., no one cut set is contained in any other cut set. More generally, with replication of basic events in the event tree, we will *not* obtain only min cut sets by this algorithm. Therefore it will be necessary, in general, to reduce the list, eliminating cut sets which contain other cut sets. The resulting list will then contain all min cut sets for the fault tree.

The cut sets obtained by the above algorithm are called *Boolean Indicated Cut Sets* (or BICS) since they will not, in general, be minimal. It is a simple matter to determine the number and maximum size of BICS for a fault tree. For large fault trees this should be done before applying the min cut algorithm in order to dimension the list matrix.

An Algorithm for Determining the Number of BICS

The number of BICS is an upper bound to the number of minimal cut sets. It is, perhaps, easiest to explain the algorithm by an example. We consider the pressure tank fault tree in Figure 7 once again. First, assign weight 1 to each of the 16 basic events. Next, assign weights to each gate starting from the bottom until we reach the top. The weight assigned to the Top Event will be the number of BICS. To an *OR* gate we assign a weight correspond to the *sum* of the weights of events input to the OR gate; thus, gates G5, G6 and G8 are each assigned weight 3 . Gate G7 is assigned weight 5 since input events 12 and 13 each have weight 1 . Gate G4 is assigned weight 8 .

To an *AND* gate we assign a weight corresponding to the *product* of the weights of the input events. Hence, gate G3 is assigned weight 24 . Gate G2 is assigned weight 26 while gate G1 is assigned weight 27 . The Top Event is assigned weight 29 . This is precisely the number of BICS found by the min cut algorithm. [See Fussell (1973).]

An Algorithm for Determining the Maximum Number of Basic Events in any BICS

As in the previous algorithm, we begin by assigning weight 1 to all basic events. However, we employ a different method of assigning weights to gates. Again, consider the pressure tank example in Figure 7. To an OR gate we assign a weight corresponding to the *maximum* of the weights of input events. Thus, gates G5 , G6 and G8 are assigned weight 1 . Likewise, gates G7 and G4 are assigned weights 1 .

To an AND gate we now assign the sum of weights corresponding to input events. Thus, gate G3 has weight 2 . Likewise, gates G2 , G1 and, finally, the Top Event have weight 2 . Recall that the maximum length of BICS obtained by our min cut algorithm for the pressure tank examples was also 2 . In general, this algorithm will only obtain an *upper bound* on the maximum size of min cut sets. [See Fussell (1973).]

See Chatterjee (1973) for a rigorous presentation and proofs of the preceding algorithms.

3. DUAL FAULT TREES

If the Top Event occurs we have system failure. This is of great interest from a safety point of view. However, from a reliability point of view, we are also interested in the nonoccurrence of the Top Event. To draw the dual fault tree, replace OR gates by AND gates and AND gates by OR gates in the original fault tree. Events are also replaced by their corresponding dual. If the Top Event is "pressure tank rupture" as in Figure 6, the dual event is "no pressure tank rupture." More generally, dual basic events correspond to the nonoccurrence of the original basic events. The dual fault tree for the pressure tank example is drawn in Figure 8.

The *min cut sets* for the dual fault tree are the *min path sets* for the original fault tree. A *path set* is a set of basic events whose nonoccurrence *insures* the nonoccurrence of the Top Event. A path set is minimal if it cannot be further reduced and still remain a path set. To find min path sets for a fault tree, draw the dual fault tree and use the min cut algorithm to find the minimal cuts for the dual fault tree. The min cut sets for the dual fault tree in Figure 8, are the min path sets for the original pressure tank fault tree of Figure 7. They are

$$\{1',2',3',4',5',6',7',8'\}$$

$$\{1',2',3',4',5',9',10',11',12',13',14',15',16'\} .$$

(We use primes to indicate dual events.) If all basic events in either of these min path sets do not occur, the Top Event in Figure 7 does *not* occur, i.e., the pressure tank does not rupture. Since there are only 2 min path sets as contrasted to 29 min cut sets, it will be easier to compute probabilities later using the min path sets.

Example. The One-out-of-Two Twice System.

The one-out-of-two twice system fault tree is presented in Figure 4. The min cut sets are {1,3} , {1,4} , {2,3} , and {2,4} . If the coolant pressure is *not* low, the occurrence of any one of the four min cut set events would produce a *spurious* alarm.

The dual of the fault tree is presented in Figure 9. This fault tree has two min cuts {1',2'} and {3',4'} . These are min paths for the original fault tree. There are thus only two min path sets in the original event tree which could cause the failure of a trip signal when low coolant pressure is actually present.

From this analysis (which neglects event probabilities) we see that the system has been designed to ensure valid alarms when low coolant pressure is present. However, it would appear prone to the production of false alarms since there are four min cut sets, any one of whose occurrence could cause a false alarm. A two-out-of-three system, for example, would be less prone to false alarms.

Relay Circuits

Yet another application of the dual fault tree concept is to relay circuits. Suppose like relays are subject to two kinds of failure: failure to close and failure to open. Similarly circuits constructed from these relays are subject to two kinds of failure: failure to close; i.e., no closed path is achieved from input wire to output wire when the circuit is commanded to close, and failure to open; i.e., a closed path exists from input wire to output wire even though the circuit is commanded to open.

If we construct a fault tree for such a circuit with Top Event-"Failure to Close", then the dual fault tree would have the dual Top Event-"Failure to Open". Thus, having constructed a fault tree for one kind of failure, the dual tree can be used to solve the second kind of failure.

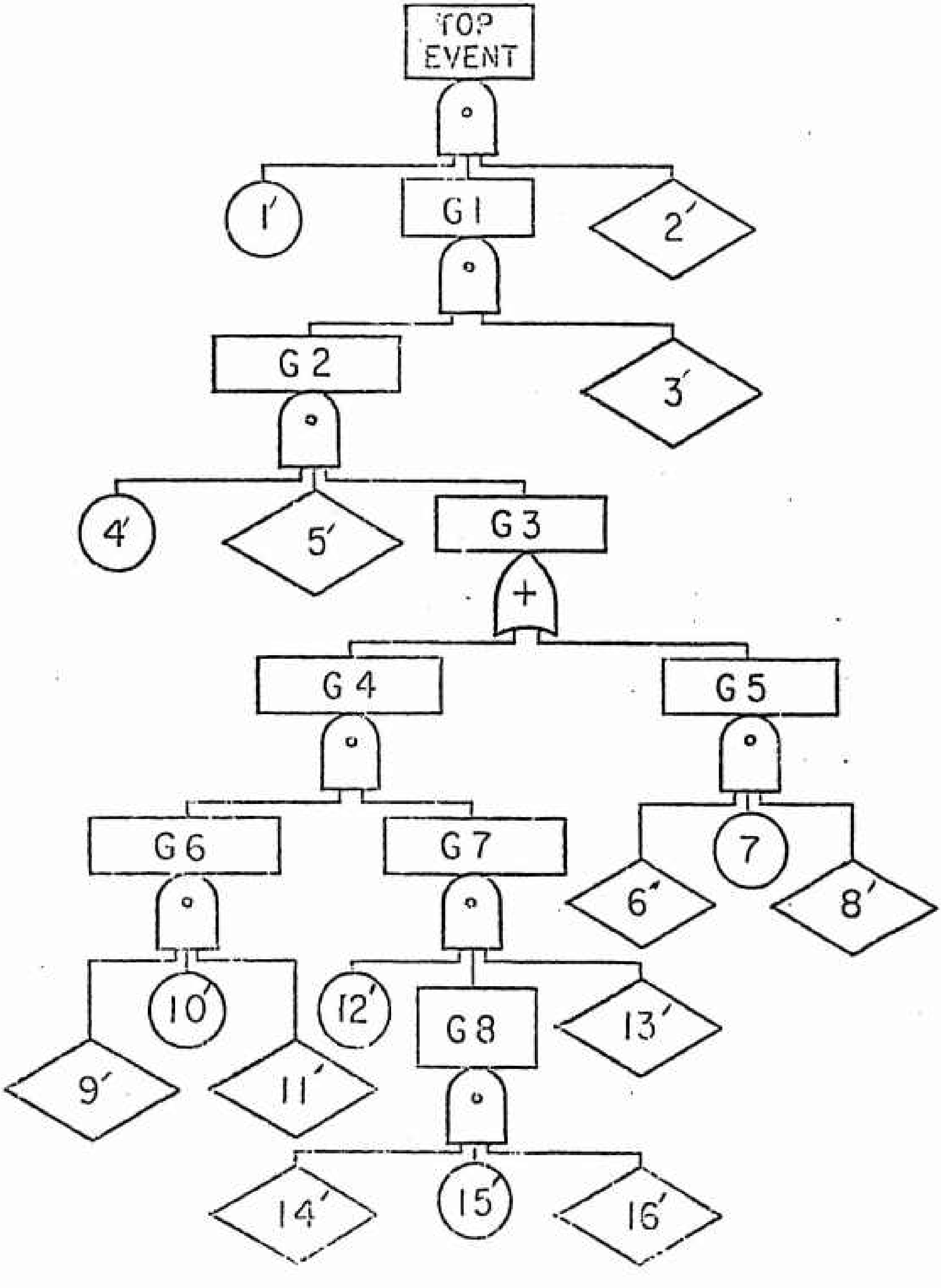


FIGURE 8: PRESSURE TANK DUAL FAULT TREE

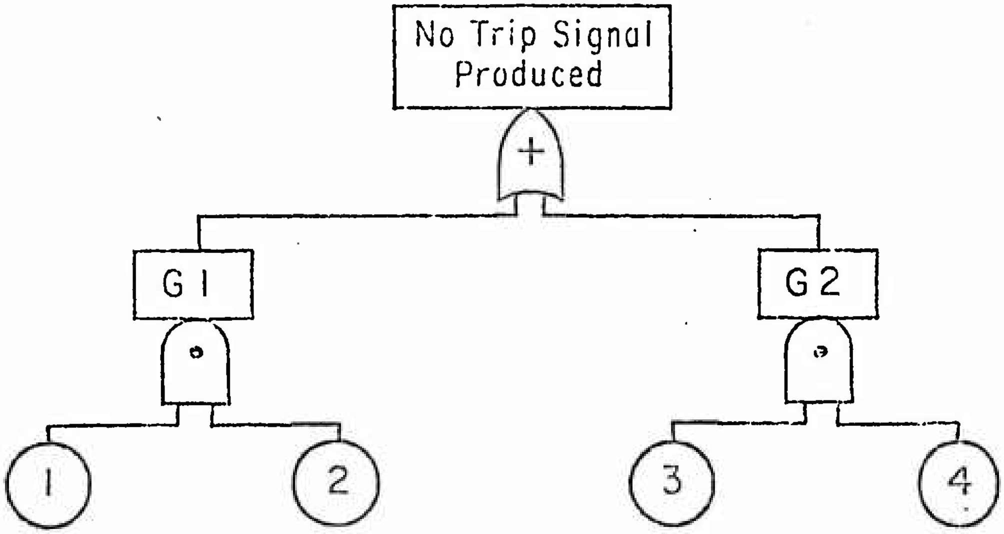


FIGURE 9: DUAL OF ONE-OUT-OF-TWO TWICE SYSTEM FAULT TREE

4. PROBABILITY EVALUATION OF FAULT TREES

A major goal of fault tree analysis is to calculate the probability of occurrence of the Top Event. However, it may also be useful to calculate the importance of min cut sets to the Top Event or the importance of specified basic events to the Top Event. We first review the most commonly used methods for calculating the probability of occurrence of the Top Event and then present new results for the case of dependent events. To make these calculations it is useful to introduce a Boolean representation for fault trees similar to that used for coherent structures [Birnbaum, Esary and Saunders (1961)].

Let

$$Y_i = \begin{cases} 1 & \text{if basic event } i \text{ occurs} \\ 0 & \text{otherwise .} \end{cases}$$

Let $\underline{Y} = (Y_1, Y_2, \ldots, Y_n)$ be the vector of basic event outcomes. Define

$$\psi(\underline{Y}) = \begin{cases} 1 & \text{if the Top Event occurs} \\ 0 & \text{otherwise .} \end{cases}$$

ψ is the Boolean indicator function for the Top Event. We assume henceforth that each basic event occurs in the union of all min cut sets; i.e. all basic events are relevant to the Top Event.

The Boolean indicator function can be determined from either the min cut sets or the min path sets. It will be convenient to introduce the notation

$$\coprod_{i=1}^{m} Y_i \overset{\text{def}}{=} 1 - \prod_{i=1}^{m} (1 - Y_i) .$$

Min Cut Representation.

Let $K_1, K_2, \ldots, K_k$ be the min cut sets of basic events for a specified fault tree. Then

$$\psi(\underline{Y}) = \coprod_{s=1}^{K} \prod_{i\in K_s} Y_i$$

is the so-called min cut representation for ψ .

Min Path Representation.

Let $P_1, P_2, \ldots, P_p$ be the min path sets of basic events for a specified fault tree. Then

$$\psi(\underline{Y}) = \prod_{r=1}^{P} \coprod_{i\in P_r} Y_i \ ,$$

is the so-called min path representation for ψ .

It is visibly obvious from either the min cut or the min path representation that ψ is coordinatewise nondecreasing.

Example. Pressure Tank System.

Let $\underline{Y} = (Y_1, Y_2, \ldots, Y_{16})$ be the random vector for basic event outcomes in the pressure tank event tree in Figure 7. Let $\psi(\underline{Y}) = 1$ if the top event occurs for outcome vector $\underline{Y}$; i.e., the pressure tank ruptures, and $\psi(\underline{Y}) = 0$ otherwise. Then using the min path sets $\{1,2,3,4,5,6,7,8\}$ and $\{1,2,3,4,5,9,10,11,12,13,14,15,16\}$ and the min path representation, we see that

$$\psi(\underline{Y}) = \left(\coprod_{1\le i\le 8} Y_i\right)\left(\coprod_{i\ne 6,7,8} Y_i\right)$$

$$= \left[1 - \prod_{i=1}^{8} (1 - Y_i)\right]\left[1 - \prod_{i\ne 6,7,8} (1 - Y_i)\right] .$$

Since there are 29 min cuts for this example, the min path representation is easier to work with.

To calculate the probability of the Top Event which in this case is pressure tank rupture, let

$$P[Y_i = 1] = EY_i = q_i$$

be the probability that basic event i occurs where E stands for expectation. For the moment, assume all basic events are statistically independent. Then

$$P[\text{Top Event}] = E\psi(\underline{Y})$$

$$(4.0) \qquad = 1 - \prod_{i=1}^{8} (1 - q_i) - \prod_{i \neq 6,7,8} (1 - q_i) + \prod_{i=1}^{16} (1 - q_i) .$$

Assume that basic event 1 (i.e., the pressure tank itself fails) occurs on the average once in 10^8 loading cycles or in other words, $q_1 = 10^{-8}$. Assume basic event i $(i \neq 1)$ occurs on the average once in 10^5 loading cycles or, in other words, $q_i = 10^{-5}$ for $i \neq 1$. Then

$$E\psi(\underline{Y}) = 1 - (1 - 10^{-8})(1 - 10^{-5})^{7} - (1 - 10^{-8})(1 - 10^{-5})^{12}$$
$$+ (1 - 10^{-8})(1 - 10^{-5})^{15} .$$

Hence

$$E\psi(\underline{Y}) \sim 4 \times 10^{-5} .$$

Boolean Reduction.

In principle we can always compute the exact probability of the top event by reducing the Boolean expression, $\psi(\underline{Y})$, for the fault tree. We do this using the fact that for Boolean variables

$$Y_i^2 = Y_i .$$

In general, once we get rid of powers of the indicator variables we can obtain the probability of the to; event by merely substituting in probabilities for indicator variables.

If there are *no* replications among min cut sets *and* basic events are *statistically independent*

$$(4.1) \qquad P[\text{TOP EVENT}] = \coprod_{1\le s\le k} \prod_{i\epsilon K_s} q_i \ .$$

If there are *no* replications among min path sets *and* basic events are *statistically independent* then

$$(4.2) \qquad P[\text{TOP EVENT}] = \prod_{1\le r\le p} \coprod_{i\epsilon P_r} q_i \ .$$

<u>Min Cut and Min Path Bounds</u>.

(4.1) and (4.2) are *not* valid in general. However, if basic events are *statistically independent*

$$(4.3) \qquad \prod_{1\le r\le p} \coprod_{i\epsilon P_r} q_i \le P[\text{TOP EVENT}] \le \coprod_{1\le s\le k} \prod_{i\epsilon K_s} q_i$$

is *always* true. The upper bound is in general quite close when the q_i's are small, which is the usual situation. (4.3) is proved in Esary and Proschan (1963).

<u>The Inclusion-Exclusion Principle</u>.

This is another method based on min cuts and can be used to obtain close bounds for large fault trees. Let E_s be the event that *all* basic events in min cut set K_s occur. We also assume *all* basic events are *statistically independent*. Then

$$P(E_s) = \prod_{i\epsilon K_s} q_i \ .$$

The top event corresponds to the event $\bigcup_{s=1}^{k} E_s$ if the fault tree has k min cut sets. Hence

$$P[\text{TOP EVENT}] = P\left[\bigcup_{s=1}^{k} E_s\right].$$

Let

$$S_r = \sum_{1 \leq i_1 < i_2 < \ldots < i_r \leq k} P\left[E_{i_1} \cap E_{i_2} \cap \ldots \cap E_{i_r}\right].$$

By the inclusion-exclusion principle

$$P[\text{TOP EVENT}] = \sum_{r=1}^{k} (-1)^{r-1} S_r$$

and

$$P[\text{TOP EVENT}] \leq S_1 = \sum_{s=1}^{k} \prod_{i \in K_s} q_i$$

$$P[\text{TOP EVENT}] \geq S_1 - S_2$$

$$P[\text{TOP EVENT}] \leq S_1 - S_2 + S_3$$

The successive upper and lower bounds, however, do not necessarily converge in a monotone fashion.

Dependent Events.

If occurrences of basic events are *not* statistically independent, then the previous methods, based on assumed independence of basic events, are no longer valid. If we know that basic events are positively dependent (the technical term we shall use is *associated*) then we can obtain useful bounds on the probability of the Top Event. First, however, we need to introduce another Boolean representation for fault trees.

The Min-Max Representation.

Let $K_1, K_2, \ldots, K_k$ be the k min cut sets for a fault tree. Then we can easily verify that

$$(4.4) \qquad \psi(\underline{Y}) = \max_{1 \leq s \leq k} \min_{i \varepsilon K_s} Y_i .$$

For, if all basic events in min cut set, say K_j, occur, then $\min_{i \varepsilon K_j} Y_i = 1$ and $\psi(\underline{Y}) = 1$, i.e. the Top Event occurs. Likewise, if $\min_{i \varepsilon K_j} Y_i = 0$ for all $1 \leq s \leq k$, then $\psi(\underline{Y}) = 0$ and the Top Event does not occur.

Sometimes it is easier to develop the fault tree structure function using the dual representation based on min paths. Let $P_1, P_2, \ldots, P_p$ be the min path sets for as specified fault tree. Then

$$\psi(\underline{Y}) = \min_{1 \leq r \leq p} \max_{i \varepsilon P_r} Y_i .$$

If $\max_{i \varepsilon P_r} Y_i = 1$ for $1 \leq r \leq p$, then a basic event occurs in each min path, i.e., the Top Event occurs so that $\psi(\underline{Y}) = 1$. If $\max_{i \varepsilon P_r} Y_i = 0$ for some r, then there is a min path set whose basic events do not occur so that the Top Event does not occur, i.e. $\psi(\underline{Y}) = 0$.

Bounds on the Probability of the Top Event.

We now assume that events are *associated:*

Definition:

[Esary, Proschan and Walkup (1967)]. Random variables $T_1, T_2, \ldots, T_n$ are associated if

$$\text{Cov}\ [\Gamma(\underline{T}), \Delta(\underline{T})] \geq 0$$

for all binary, increasing functions Γ and Δ .

In a great many reliability situations, the random variables of interest are *not* independent, but rather are "associated". As examples, consider

(a) indicator functions of min cut sets which have basic events in common;

(b) components subjected to a common environment;

(c) structures in which components share the load, so that failure of one component results in increased load on each of the remaining components.

In case (a), if the basic events are independent, the min cut indicator functions are associated and not independent. Examples (b) and (c) are physical situations which could lead to associated indicator random variables.

Theorem 4.1:

If indicator random variables $Y_1, Y_2, \ldots, Y_n$ are associated, then

$$(4.5) \qquad \max_{1 \leq s \leq k} \prod_{i \varepsilon K_s} q_i \leq P[\text{Top Event}] \leq \min_{1 \leq r \leq p} \coprod_{i \varepsilon P_r} q_i \ .$$

Note that, in contrast to (4.3), the lower bound depends on min cut sets.

Proof:

The following always holds

$$\min_{i \varepsilon K_s} Y_i \leq \psi(\underline{Y}) \leq \max_{i \varepsilon P_r} Y_i$$

for all r $(1 \leq r \leq p)$ and s $(1 \leq s \leq k)$. It follows that

$$\max_{1 \leq s \leq k} P[\min_{i \varepsilon K_s} Y_i = 1] \leq P[\psi(\underline{Y}) = 1] \leq \min_{1 \leq r \leq p} P[\max_{i \varepsilon P_r} Y_i = 1]$$

is always true.

Since $Y_1, Y_2, \ldots, Y_n$ are associated

(4.6)
$$E \prod_{i \varepsilon K_s} Y_i \geq \prod_{i \varepsilon K_s} q_i$$

and

(4.7)
$$E \coprod_{i \varepsilon P_r} Y_i \leq \coprod_{i \varepsilon P_r} q_i$$

[Esary, Marshall and Proschan (1967)]. (4.5) follows from the observation that

$$\min_{i \varepsilon K_s} Y_i = \prod_{i \varepsilon K_s} Y_i$$

and

$$\max_{i \varepsilon P_r} Y_i = \coprod_{i \varepsilon P_r} Y_i \ . ||$$

If basic events are statistically independent and the q_i's are small, the upper bound in (4.3) will very likely be the better bound. However, for large values of the q_i's , (4.5) may provide the better bound. To illustrate this, consider a fault tree with min cut sets

$K_1 = \{1,2\}$, $K_2 = \{1,3\}$, $K_3 = \{1,4\}$, $K_4 = \{2,3\}$, $K_5 = \{2,4\}$, $K_6 = \{3,4\}$.

For simplicity suppose $q_1 = q_2 = q_3 = q_4 = q$. The upper bound in (4.3) is $1 - [1 - q^2]^6$ while the upper bound in (4.5) is $1 - (1 - q)^3$. The min-max upper bound is smaller than the min cut upper bound when $q \geq .62$.

<u>Example: The Pressure Tank</u>.

Assume $q_1 = 10^{-8}$ and $q_2 = q_3 = \ldots = q_{16} = 10^{-5}$. Then

$$P[\psi(\underline{Y}) = 1] \leq \min_{1 \leq r \leq 2} \coprod_{i \varepsilon P_r} q_i \sim 7 \times 10^{-5} \ .$$

On the other hand

$$P[\psi(\underline{Y}) = 1] \geq \max_{1 \leq s \leq 29} \prod_{i \in K_s} q_i \sim 10^{-5} .$$

Hence, assuming only that basic events are associated, we have

$$10^{-5} \leq P[\text{Top Event}] \leq 7 \times 10^{-5} .$$

Modules.

A module of a fault tree is a set of basic events M, together with an indicator function χ_M, such that

$$\psi(\underline{Y}) = \Gamma\left[\chi_M(\underline{Y}^M), \underline{Y}^{M'}\right]$$

where Γ is nondecreasing and $\underline{Y}^M$ means the coordinates of $\underline{Y}$ are restricted to M. Modules were described for coherent structures by Birnbaum and Esary (1965). Decomposing a tree in terms of modules can be useful in reducing the computation required for probabilistic evaluation of fault trees. Suppose we can find a modular decomposition $\{(M_1,\chi_1), \ldots, (M_r,\chi_r)\}$ such that $\chi_1(\underline{Y}), \ldots, \chi_r(\underline{Y})$ are statistically independent, although Y_i for $i \in M_s$ $(1 \leq s \leq r)$ may be associated. Then

$$\begin{aligned} P[\psi(\underline{Y}) = 1] &= g_\Gamma[P[\chi_1(\underline{Y}) = 1], \ldots, P[\chi_r(\underline{Y}) = 1]] \\ &\leq g_\Gamma\left[u_{\chi_1}(\underline{q}), \ldots, u_{\chi_r}(\underline{q})\right] \end{aligned} \tag{4.8}$$

where $u_{\chi_s}(\underline{q})$ is the min-max upper bound, (4.5), for module M_s and g_Γ is the expected value of $\Gamma[\chi_1, \ldots, \chi_r]$. (4.8) follows from the monotonicity of g_Γ. In applications, it may be useful to decompose the tree into statistically independent modules and apply (4.8) rather than to apply (4.5) directly since (4.5) will be more conservative.

Time to Occurrence of the Top Event.

First, we suppose that once a basic event occurs, it cannot be rectified. Suppose basic event i occurs at time T_i and the Top Event occurs at time T. Let

$$Y_i(t) = \begin{cases} 1 & \text{if } T_i \leq t \\ 0 & \text{otherwise.} \end{cases}$$

Then P[Top Event occurs by time t]

$$= E\psi[\underline{Y}(t)]$$

where $\underline{Y}(t) = (Y_1(t), \ldots, Y_n(t))$, since ψ is nondecreasing. If $P[T_i \leq t] = F_i(t)$ then we can compute $E\psi[\underline{Y}(t)]$ by using the previous algorithms with q_i replaced by $F_i(t)$. In particular, (4.5) becomes

$$\max_{1 \leq s \leq k} \prod_{i \varepsilon K_s} F_i(t) \leq E\psi[\underline{Y}(t)] \leq \min_{1 \leq r \leq p} \coprod_{i \varepsilon P_r} F_i(t) .$$

Mean Time to Occurrence of the Top Event.

To calculate the mean time to occurrence of the Top Event we need the distribution of time to occurrence of the Top Event. Since this is often difficult or impossible to compute, we obtain a useful lower bound on the mean.

First, we observe that

$$T = \min_{1 \leq s \leq k} \max_{i \varepsilon K_s} T_i \tag{4.9}$$

and also

$$T = \max_{1 \leq r \leq p} \min_{i \varepsilon P_r} T_i . \tag{4.10}$$

To see (4.9) note that the Top Event occurs as soon as the first min cut event occurs. A specified min cut can only cause the Top Event after the last event time in the min cut set.

To see (4.10) note that the Top Event occurs *after* the last min path fails. A min path set fails as soon as any event in the set occurs.

To obtain the mean time to occurrence of the Top Event, ET , one might think of substituting mean occurrence times in (4.9) or (4.10). This will *not* give the expected time to occurrence of the Top Event.

Theorem 4.2:

If times to occurrence of basic events are associated and $\frac{-\log \bar{F}_i(t)}{t}$ is nondecreasing in $t \ge 0$ for $i = 1,2, \ldots, n$, then

$$(4.11) \qquad \max_{1\le r\le p} \left[\sum_{i\varepsilon P_r} \mu_i^{-1}\right]^{-1} \le ET \le \min_{1\le s\le k} \int_0^\infty \prod_{i\varepsilon K_s} \bar{G}_i(t)dt$$

where $\mu_i = \int_0^\infty t dF_i(t)$ and $\bar{G}_i(t) = e^{-t/\mu_i}$ for $i = 1,2, \ldots, n$. (If F_i has nondecreasing occurrence rate, $dF_i(t)/\bar{F}_i(t)$, then $\frac{-\log \bar{F}_i(t)}{t}$ is nondecreasing for $t \ge 0$.)

Proof:

Using (4.9) and (4.10) we see that

$$E \min_{i\varepsilon P_r} T_i \le ET \le E \max_{i\varepsilon K_s} T_i$$

holds for $1 \le r \le p$ and $1 \le s \le k$. Hence

$$(4.12) \qquad \max_{1\le r\le p} E \min_{i\varepsilon P_r} T_i \le ET \le \min_{1\le s\le k} E \max_{i\varepsilon K_s} T_i \; .$$

To show the upper bound, observe that

$$P[\max_{i\varepsilon K_s} T_i > t] = P[\prod_{i\varepsilon K_s} [1 - Y_i(t)] = 1]$$

$$\le \prod_{i\varepsilon K_s} P[Y_i(t) = 0]$$

by association [Esary, Proschan and Walkup (1967)]. Also

$$E \max_{i\varepsilon K_s} T_i = \int_0^\infty P[\max_{i\varepsilon K_s} T_i > t]dt$$

$$\leq \int_0^\infty \prod_{i\varepsilon K_s} P[Y_i(t) = 0]dt \leq \int_0^\infty \prod_{i\varepsilon K_s} \overline{G}_i(t)dt$$

since $F_i(t) = P[Y_i(t) = 1]$, $i = 1,2, \ldots, n$ have the property that $\dfrac{-\log \overline{F}_i(t)}{t}$ is nondecreasing, i.e. F is IFRA for increasing failure rates on the average. [Marshall and Proschan (1970)]. The upper bound follows by substituting in (4.12).

To show the lower bound, observe that

$$P[\min_{i\varepsilon P_r} T_i > t] = P[\prod_{i\varepsilon P_r} (1 - Y_i(t)) = 1]$$

$$\geq \prod_{i\varepsilon P_r} P[Y_i(t) = 0]$$

by association [Esary, Proschan and Walkup]. Also

$$E \min_{i\varepsilon P_r} T_i = \int_0^\infty P[\min_{i\varepsilon P_r} T_i > t]dt$$

$$\geq \int_0^\infty \prod_{i\varepsilon P_r} P[Y_i(t) = 0]dt \geq \int_0^\infty \prod_{i\varepsilon P_r} \overline{G}_i(t)dt$$

$$= \left[\sum_{i\varepsilon P_r} \mu_i^{-1}\right]^{-1}$$

again using the IFRA property of F_i $(i = 1,2, \ldots, n)$. The lower bound follows by substituting in (4.12).||

Example. The Pressure Tank.

Suppose $ET_1 = 10^8$ cycles and $ET_i = 10^5$ cycles for $i > 1$. Then, using the min path sets

$$ET \geq \max\,[14{,}283,\ 8{,}332] = 14{,}283 \text{ cycles.}$$

5. MEASURES OF EVENT IMPORTANCE

The next step after obtaining the fault tree minimal cuts is to determine the relative importance of basic events to the occurrence of the Top Event. From the list of min cuts for the pressure tank example, it is intuitively clear that basic events 1 , 2 , 3 , 4 and 5 are the most important since each is a one component min cut. However, the relative importance of the remaining basic events is less clear.

Suppose the Top Event occurs and we perform an autopsy to determine the cause. In practice we may find that several min cuts have occurred. However, if we think of events occurring sequentially in time and suppose two or more events cannot occur precisely at the same instant, then there must have been one event which "caused" the Top Event.

In order to compute the probability that basic event i causes the Top Event, let $F_i(t)$ be the probability that basic event i $(i = 1,2, \ldots, n)$ occurs before time t . We also assume F_i continuous. Let $p_i = 1 - q_i$ and

$$h(\underline{p}) = 1 - E\psi(\underline{Y})$$

be the probability that the Top Event does *not* occur where $\underline{p} = (p_1, p_2, \ldots, p_n)$. If all basic events have the same occurrence distribution (or have approximately equal occurrence rates) then it is shown in Barlow and Proschan (1973) that

$$(5.1) \qquad \int_0^1 [h(1_i,p) - h(0_i,p)]dp$$

is the probability that basic event i causes the Top Event, where

$$h(1_i,p) = h(p, \ldots, p, 1_i, p, \ldots, p)$$

and

$$h(0_i,p) = h(p, \ldots, p, 0_i, p, \ldots, p) .$$

Example: The Pressure Tank (equal occurrence rates)

From (4.0) we see that

$$h(\underline{p}) = \prod_{i=1}^{8} p_i + \prod_{i\neq 6,7,8} p_i - \prod_{i=1}^{16} p_i .$$

For $i = 1$,

$$h(1_i,p) = p^7 + p^{12} - p^{15}$$

and

$$h(0_i,p) = 0 .$$

Hence

$$\int_0^1 [h(1_i,p) - h(0_i,p)]dp = .13942 .$$

Assuming all events have equal occurrence rates, the likelihood that the pressure tank causes the Top Event is approximately .14 .

More generally, let E_i be the event that basic event i causes the Top Event. Then

$$P[E_1] = \ldots = P[E_5] = .13942$$

$$P[E_6] = P[E_7] = P[E_8] = .0625$$

$$P[E_9] = \ldots = P[E_{16}] = .01442 .$$

Note that the probabilities sum to one as they should, since when the Top Event occurs, it must have been caused by one of events 1 through 16.

Events 1 through 5 will cause 70% of the failures in this case. Note that it was unnecessary to know the common occurrence rate.

Proportional Occurrence Rates.

We say that event occurrence distributions, F_i , have proportional occurrence rates if

$$\overline{F}_i(t) = [\,\overline{F}(t)]^{\lambda_i}$$

where $\lambda_i > 0$, $i = 1,2, \ldots, n$. It is only necessary to specify the λ_i's to compute the probability that basic event i causes the Top Event. The computing formula is

$$(5.2) \qquad \int_0^1 \left[h\left(1_i, p^{\underline{\lambda}}\right) - h\left(0_i, p^{\underline{\lambda}}\right)\right] \lambda_i p^{\lambda_i - 1}\, dp$$

where

$$\left(1_i, p^{\underline{\lambda}}\right) = \left(p^{\lambda_1}, \ldots, p^{\lambda_{i-1}}, 1, p^{\lambda_{i+1}}, \ldots, p^{\lambda_n}\right)$$

and

$$\left(0_i, p^{\underline{\lambda}}\right) = \left(p^{\lambda_1}, \ldots, p^{\lambda_{i-1}}, 0, p^{\lambda_{i+1}}, \ldots, p^{\lambda_n}\right) .$$

(5.2) is proved in Barlow and Proschan (1973).

Example. The Pressure Tank. (unequal occurrence rates)

Assuming basic event 1 has occurrence rate 10^{-8} per cycle while all other events have occurrence rate 10^{-5} per cycle, we wish to calculate the probability of basic event i causing the Top Event. In this case $\lambda_1 = 10^{-3}\lambda_i$ for $i > 1$. For convenience, let $\lambda_1 = .001$ and $\lambda_i = 1$ for $i > 1$. (Actually occurrence rates could be time dependent so long as the proportions are as assumed.) Using (5.2) we calculate

$$P[E_1] = .0001595$$

$$P[E_2] = P[E_3] = P[E_4] = P[E_5] = .1595$$

$$P[E_6] = P[E_7] = P[E_8] = .07617$$

$$P[E_9] = \ldots = P[E_{16}] = .016664$$

The importance of events 2 through 8 have increased by about 1% over the previous example while event 1 is now negligible.

The importance of min cut sets is discussed in Barlow and Proschan (1973).

Marginal Importance of Basic Events.

Birnbaum (1969) proposed $\frac{\partial h(\underline{p})}{\partial p_i}$ as a measure of the importance of basic event i. This measure of event importance is useful for determining design improvements based on cost considerations. Letting $p_1 = p_2 = \ldots = p_n = 1/2$, be called this, the structural (marginal) importance of basic event i. This can also be described in terms of critical path sets.

C_i is a *critical path set for basic event* i if it is a path set containing i such that each of its min path sets contains i. Let $n(i)$ be the number of *critical path sets for* i. Then we define the Birnbaum *importance of basic event* i by

$$B(i) = 2^{-(n-1)} n(i) ,$$

where n denotes the number of basic events in the event tree.

To compute $n(i)$, assume the Y_i's are statistically independent, $EY_i = E(1 - Y_i) = 1/2$ for $i = 1,2, \ldots, n$, and use the formula

$$n(i) = 2^{n-1} E[\psi(1_i,\underline{Y}) - \psi(0_i,\underline{Y})]$$

[Cf. Barlow and Proschan (1973).]

Example. The Pressure Tank.

Use $\psi(\underline{Y}) = \left[1 - \prod_{i=1}^{8} (1 - Y_i)\right]\left[1 - \prod_{i\neq 6,7,8} (1 - Y_i)\right]$ to compute

$$E[\psi(1_j,\underline{Y}) \mid EY_i = 1/2, i = 1,2, \ldots, n]$$

and

$$E[\psi(0_j,\underline{Y}) \mid EY_i = 1/2, i = 1,2, \ldots, n] .$$

For basic event 1 ,

$$n(1) = 2^{15}\left\{1 - E\left[1 - \prod_{\substack{i=1 \\ i\neq 1}}^{8} (1 - Y_i)\right]\left[1 - \prod_{i\neq 1,6,7,8} (1 - Y_i)\right]\right\}$$

$$= 2^{15}[(1/2)^7 + (1/2)^{12} - (1/2)^{15}] = 263 .$$

It is not hard to see that

$$n(1) = n(2) = n(3) = n(4) = n(5) = 263 .$$

For basic event 6, $n(6) = 255$.

Also $n(6) = n(7) = n(8) = 255$.

For basic event 9, $n(9) = 7$. It is not hard to see that

$$n(9) = n(10) = n(11) = n(12) = n(13) = n(14) = n(15) = n(16) = 7 .$$

The Birnbaum importance ordering of events is therefore

$$1 \sim 2 \sim 3 \sim 4 \sim 5 > 6 \sim 7 \sim 8 > 9 \sim 10 \sim 11 \sim 12 \sim 13 \sim 14 \sim 15 \sim 16 ,$$

where "$1 \sim 2$" means 1 and 2 are equally important in the event tree, and "$5 > 6$" means 5 is more important than 6 in the event tree. Figure 10 provides a key to the original example of Figure 6. For example, we see that the pressure tank itself and the K2 relay are structurally most important. The pressure switch is next most important, while the timer, the K1 relay, and the S1 switch are the least important structurally.

Basic Event i	Prob. i causes rupture	Number of Critical Paths, n(i), Containing Basic Event i	Description of Basic Events
1	(.000159)	263	Pressure tank failure
2	(.159500)	263	Secondary failure of pressure tank due to improper selection
3	(.159500)	263	Secondary failure of pressure tank to out-of-tolerance conditions
4	(.159500)	263	K2 Relay contacts fail to open
5	(.159500)	263	K2 Relay secondary failure
6	(.0761745)	255	Pressure switch secondary failure
7	(.0761745)	255	Pressure switch contacts fail to open
8	(.0761745)	255	Excess pressure not sensed by pressure actuated switch
9	(.016664)	7	S1 switch secondary failure
10	(.016664)	7	S1 switch contacts fail to open
11	(.016664)	7	External reset actuation force remains on switch S1
12	(.016664)	7	K1 relay contacts fail to open
13	(.016664)	7	K1 relay secondary failure
14	(.016664)	7	Timer does not "time off" due to improper setting
15	(.016664)	7	Timer relay contacts fail to open
16	(.016664)	7	Timer relay secondary failure

FIGURE 10: KEY TO PRESSURE TANK EXAMPLE

6. COMPUTER PROCESSING OF FAULT TREES

In this section we give a brief description of a Fortran program called TREEL which has been developed for processing fault trees.

The handling of complex systems necessitates various error checks on the input data. Fault trees are represented to the computer by describing each gate of the tree with one card. It contains an alpha-numeric name of the gate, type of the gate, number of gate inputs and basic event input and their alphanumeric names. The program 'TREEL' not only makes error checks from punching mistakes to circular logic, but also reindexes the gates and components. The importance of this indexing is tremendous in analyzing the fault tree in an efficient manner.

For a system with 2000 gates and 2000 basic events we would index the basic events from 1 to 2000 and gates by integers from 2001 to 4000. Gates are indexed in the order they appear in the tree from the bottom, i.e. the lowest level gates are those which have only basic events as inputs. This indexing scheme assures us that if a gate gets index I then it has inputs whose indices will be less than I.

Apart from indexing the gate, it also produces the Fortran equivalent of the tree logic. Thus we can evaluate the system state given the component states.

We also obtain bounds on the number of min cut sets and max size of the min cut sets of this tree as well as the dual tree. This information is a valuable aid in determining which tree to work on.

We also obtain the degree of replication of the gates and basic events in the tree. The number of times a gate is replicated in the tree is a helpful aid in reducing storage requirements of min cut set algorithms [Chatterjee (1973)].

Subroutine XREF prints out the cross-reference table of the tree index and the alphanumeric identification names of the gates and basic events.

The program is written in FORTRAN for the CDC 6400. This program has lower storage requirements, shorter execution time and more flexibility (i.e. is not just restricted to 'AND' and 'OR' gates) than the comparable program of Veseley and Narum [1970]. The generalized version of the program takes care of any gate for which the logic function is well defined and can be written as a FUNCTION routine.

REFERENCES

Barlow, R. E. and F. Proschan, "Time Dependent Importance of Components and Events," Operations Research Center Technical Report, University of California, Berkeley, California, (1973).

Birnbaum, Z. W., J. D. Esary and S. C. Saunders, "Multicomponent Systems and Structures, and their Reliability," Technometrics, Vol. 3, pp. 55-77, (1961).

Birnbaum, Z. W. and J. D. Esary, "Modules of Coherent Binary Systems," SIAM Journal on Applied Math., Vol. 13, pp. 444-462, (1965).

Birnbaum, Z. W., "On the Importance of Different Components in a Multicomponent System," in MULTIVARIATE ANALYSIS-II, edited by P. R. Krishnaiah, Academic Press, (1969).

Chatterjee, P., "Fault Tree Analysis: Min Cut Set Algorithms," to appear as ORC Technical Report, Operations Research Center, University of California, Berkeley, (1973).

Esary J. D., F. Proschan and D. W. Walkup, "Association of Random Variables, with Applications," Ann. Math. Statist., Vol. 38, pp. 1466-1474, (1967).

Fussell, J. B. and W. E. Vesely, "A New Methodology for Obtaining Cut Sets for Fault Trees," American Nuclear Society Transactions, Vol. 15, No. 1, pp. 262-263, (1972).

Fussell, J. B., "Fault Tree Analysis--Concepts and Techniques," Aerojet Nuclear Company Report, Idaho Falls, Idaho, (1973).

Haasl, D., "Advanced Concepts in Fault Tree Analysis," System Safety Symposium, Sponsored by the University of Washington and the Boeing Company, Seattle, Washington, (1965).

Lambert, H., "Systems Safety Analysis and Fault Tree Analysis," Lawrence Livermore Laboratory Report UCID-16238, Livermore, Calif., (1973).

Marshall, A. W. and F. Proschan, "Mean Life of Series and Parallel Systems," J. Appl. Prob., Vol. 7, pp. 165-174, (1970).

Vesely, W. E., "A Time-Dependent Methodology for Fault Tree Evaluation," Nuclear Engineering and Design, Vol. 13, No. 2, (1970).

Esary, J. D. and F. Proschan, "Coherent Structures of Non-Identical Components," Technometrics, Vol. 5, pp. 191-209, (1963).